Pharmacy Managemant system Data Base

Our Team

Mostafa
Hammam

Sayed
Abozaid

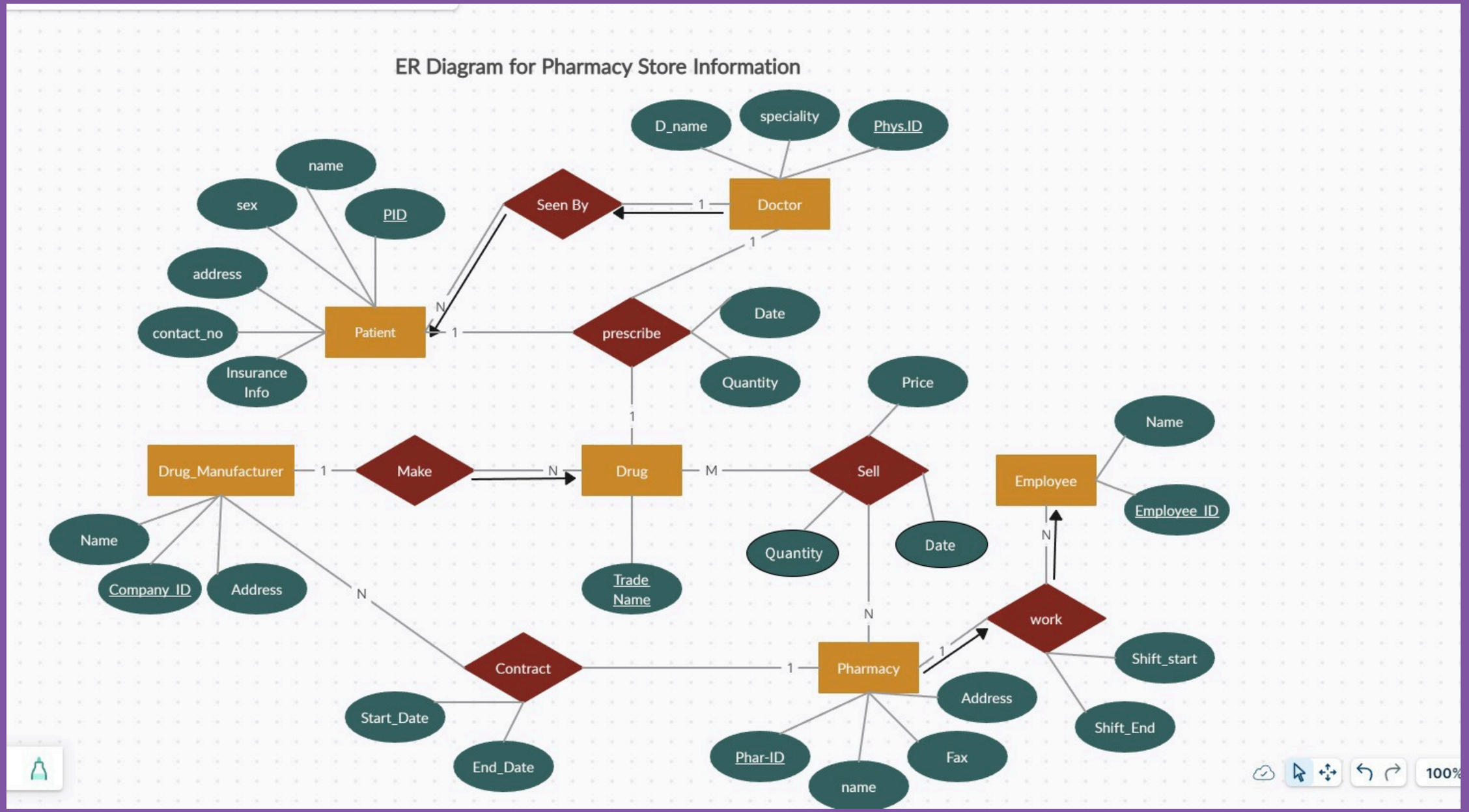
Ahmed
Hamidou

Mohammed
Ahmed

OVERVIEW

INTRODUCTION

The objective of this project is to design and implement a SQL database for managing and querying data related to a group of pharmacies. The database will store information about sales, pharmacies, employees, contracts, patients, doctors, prescriptions, drugs, and drug manufacturers. The primary goal is to facilitate efficient business queries that provide insights into the pharmacy operations. Additionally, the project will focus on ensuring the security of sensitive data and optimizing the efficiency of the SQL code.


ER Diagram for Pharmacy Store Information
D_name
speciality
Phys.ID
Doctor
Seen By
1
1
name
sex
PID
address
contact_no
Insurance Info
Patient
N
1
prescribe
Date
Quantity
1
Price
Drug_Manufacturer
1
Make
N
Drug
M
Sell
Name
Company_ID
Address
N
Trade Name
Quantity
Date
Employee
Name
Employee_ID
N
work
N
Contract
1
Pharmacy
1
Shift_start
Shift_End
Address
Start_Date
End_Date
Phar-ID
name
Fax
100%

MAPPING

Sale(saleid,drugid,pharid,date,quantity,price)

Pharmacy(pharid, name, address, fax)

Employee(eid, name, pharid, shiftstart, shiftend)

Drug(drugid, tradename, companyid, quantity)

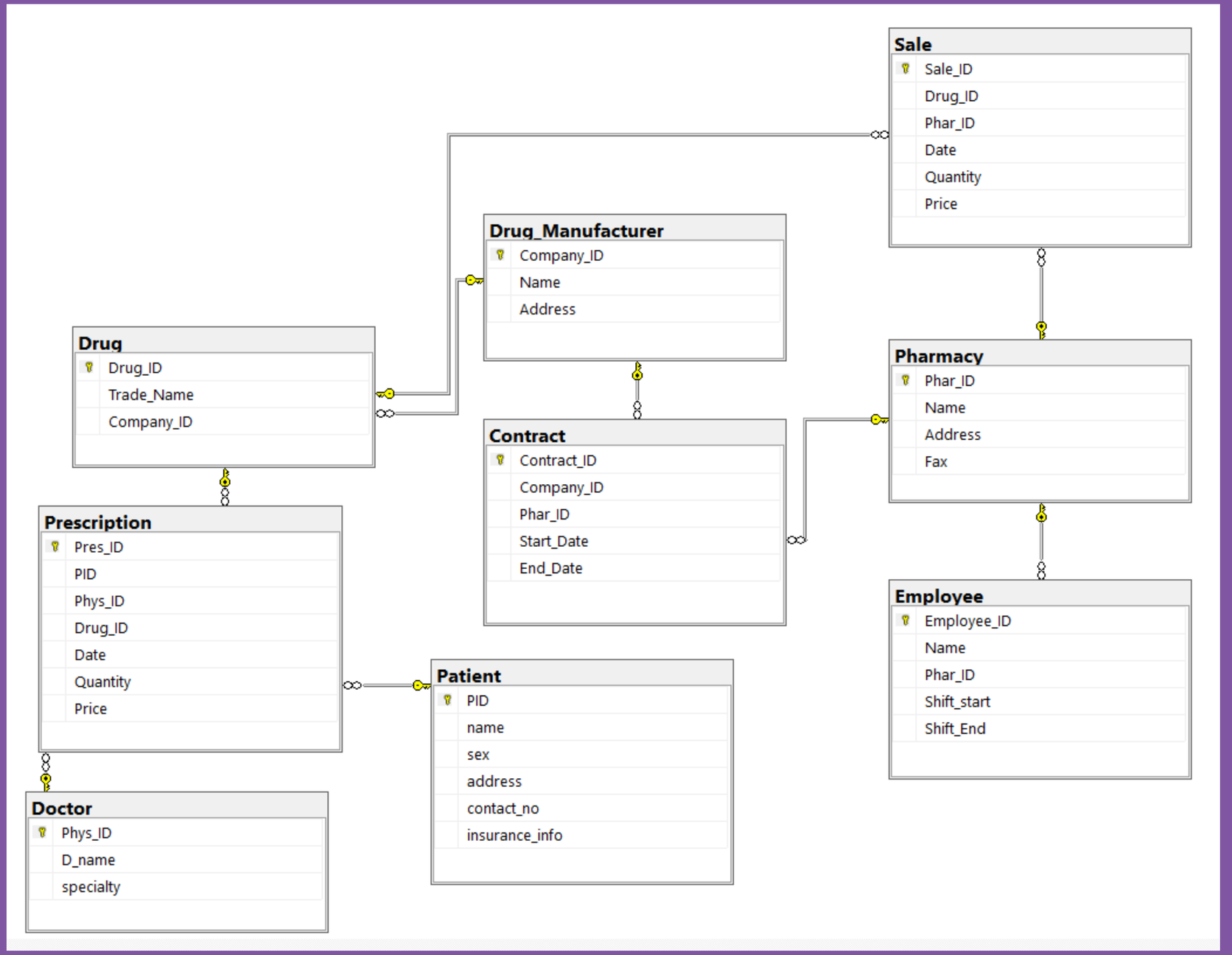
Drug_Manufacturer(companyid, name, address)

Contract(contractid, companyid, pharid, startdate, enddate)

Doctor(physid, dname, specialty)

Patient(pid, name, sex, address, contactno, insuranceinfo)

Prescription(presid, pid, physid, drugid, date, quantity, price)


Sale
Sale_ID
Drug_ID
Phar_ID
Date
Quantity
Price
Drug_Manufacturer
Company_ID
Name
Address
Drug
Drug_ID
Trade_Name
Company_ID
Pharmacy
Phar_ID
Name
Address
Fax
Contract
Contract_ID
Company_ID
Phar_ID
Start_Date
End_Date
Prescription
Pres_ID
PID
Phys_ID
Drug_ID
Date
Quantity
Price
Employee
Employee_ID
Name
Phar_ID
Shift_start
Shift_End
Patient
PID
name
sex
address
contact_no
insurance_info
Doctor
Phys_ID
D_name
specialty

```
--------------------------------------------------------------------------------
--1.Query using Cursor to view details of patients with specific prescriptions.
declare details_of_patients cursor
for
    select p.PID,name, Pres_ID
    from Patient p join Prescription psc
    on p.PID=psc.PID
    order by p.PID

    declare @pid int, @name varchar (50), @pres_id int

open details_of_patients
        fetch details_of_patients into @pid , @name , @pres_id
        while @@FETCH_STATUS=0
begin
        select @pid as[P_ID], @name as[Name], @pres_id as[Pres_ID]
        fetch details_of_patients into @pid , @name , @pres_id
end
close details_of_patients
deallocate details_of_patients
```

```
--4.  A query using Cursor to display details of medications that have been prescribed in more than one pharmacy
declare c4 cursor
for
select d.Drug_ID, d.Trade_Name, d.Company_ID
from Drug d inner join Sale s
on s.Drug_ID=d.Drug_ID
group by d.Drug_ID, d.Company_ID, d.Trade_Name
having count(distinct s.Phar_ID) > 1
for read only

declare @drugid int, @trade_name varchar(30), @companyid int
open c4
fetch c4 into @drugid , @trade_name, @companyid

while @@FETCH_STATUS=0
begin
select @drugid , @trade_name, @companyid
fetch c4 into @drugid , @trade_name, @companyid
end

close c4
deallocate c4
```

```
--3- Query using Cursor to view details of doctors who have more than 5 patientsdeclare doc_with_more_than_4_patients
for
    select distinct d.Phys_ID,d.D_name,count(p.PID)over(partition by p.phys_id order by p.phys_id) as nop
    from Prescription p , Doctor d
    where p.Phys_ID = d.Phys_ID

for READ ONLY
declare @Phys_ID varchar(10) , @D_name varchar(50),@nop int , @doc_names_with_more_than_4 varchar(50)
open doc_with_more_than_4_patients

fetch doc_with_more_than_4_patients into  @Phys_ID,@D_name , @nop
while @@FETCH_STATUS=0
begin
if  @nop > 4
set @doc_names_with_more_than_4 = @D_name

   fetch doc_with_more_than_4_patients into @Phys_ID,@D_name , @nop
end
select @doc_names_with_more_than_4
close doc_with_more_than_4_patients
deallocate doc_with_more_than_4_patients
```

```
 Cursor is used to display details of medications whose price exceeds the average
rugID INT, @TradeName VARCHAR(100), @Price FLOAT;
ugCursor CURSOR FOR
g_ID, Trade_Name, Price

e > (SELECT AVG(Price) FROM Drug);

ursor;

 FROM DrugCursor INTO @DrugID, @TradeName, @Price;

TCH_STATUS = 0

ug ID: ' + CAST(@DrugID AS VARCHAR(10))+', Name: ' + @TradeName +', Price: '+ CAST(@Price AS VAR


Cursor;
 DrugCursor;
```

```sql
--3. Create a Function to calculate the amount of medication prescribed to a specific patient

create or alter function nofprescripedmidicines(@pid int)
returns int
  begin
     declare @num int
     select @num = count( pr.Drug_ID)
     from  Prescription pr
     where pr.PID=@pid


     RETURN @num
  end

 select dbo.nofprescripedmidicines(2)
```

```sql
--2. Create a Function to count the number of prescriptions for a specific patient

CREATE FUNCTION dbo.PatientPrescriptionCount(@PatientID INT)
RETURNS INT
AS
BEGIN
    DECLARE @Count INT;
    SELECT @Count = COUNT(*)
    FROM Prescription
    WHERE PID = @PatientID;
    RETURN @Count;
END;
SELECT dbo.PatientPrescriptionCount(1) AS PrescriptionCount;
```

```sql
.Create a function to calculate the average prices of medicines in a given prescriptior
ate or alter function get_avg_price (@pres_id int)
URNS int
IN

lare @Avg_Price int
 select @Avg_Price = avg(price)
 from Prescription
 where @Pres_ID=Pres_ID
urn @Avg_Price


ect dbo.get_avg_price (45) as [Avg Price]
```

```sql
--4. Create a Function to calculate the total prescription costs for a given patient

create function pres_price(@patient int)
returns int
begin
declare @total_price int
select @total_price = sum(Price * quantity)
from Prescription
where PID = @patient
return @total_price
end

select dbo.pres_price(2)
```

```sql
--3- Create a DML Stored Procedure to delete a patient based on the patient ID
create or alter proc delete_patient @dpid int
as
delete from  prescription
where PID=@dpid
delete from patient
where PID=@dpid
execute delete_patient 130


--4- Create a DML Stored Procedure to add a new recipe
CREATE PROC add_newrow
    @Pres INT,
    @pid INT,
    @phys INT,
    @drug INT,
    @Date DATE,
    @quantity int,
    @price decimal
AS
BEGIN
    INSERT INTO Prescription (Pres_ID, PID, Phys_ID, Drug_ID, Date, Quantity, Price)
    VALUES (@Pres, @pid, @phys, @drug, @Date, @quantity, @price)
END

execute add_newrow 101, 20, 10, 40, '2025-04-01', 44, 36
```

```sql
-------------------------------------------------------------------------
--1.Create a DML Stored Procedure to Add a New Patient.
create or alter proc sp_New_patient
        @pid int, @name varchar(50), @sex varchar (1), @address nvarchar(50),
        @contact_no nvarchar(15), @insurance_info nvarchar(255)
as
begin
insert into Patient (pid, name, sex, address, contact_no, insurance_info)
            values(@pid, @name, @sex, @address, @contact_no, @insurance_info);
end
go
execute sp_New_patient @pid=131 ,@name='Ahmed', @sex='M', @address='9898_Sohag' ,
        @contact_no='555_6666', @insurance_info='Insurance R5';


--2- Create a DML Stored Procedure to update the details of a specific drug
CREATE PROCEDURE UpdateDrugDetails
    @DrugID INT,
    @TradeName VARCHAR(100),
    @companyid int,
    @Stock INT
AS
BEGIN
    UPDATE Drug
    SET Trade_Name = @TradeName, Price = @Price, Stock = @Stock
    WHERE Drug_ID = @DrugID;
END;

EXEC UpdateDrugDetails 1, 'New Drug Name5', 50.0, 200;
```

```sql
--------------------------------------------------------------------------------
--1.A query that uses Ranking to rank doctors based on the number of patients.
select d_name , count(d.Phys_ID) as [Number of Patients],
row_number() over(order by count(d.phys_id)desc) as [doctor rank], d.Phys_ID
from doctor d join Prescription psc on d.Phys_ID=psc.Phys_ID
              join Patient p on p.PID=psc.PID
group by D_name, d.Phys_ID
ORDER BY [Number of Patients] DESC;

--2- A query that uses Ranking to classify patients based on the number of prescriptions

SELECT name,
       RANK() OVER (ORDER BY COUNT(Prescription.Pres_ID) DESC) AS Rank
FROM Patient
JOIN Prescription ON Patient.PID = Prescription.PID
GROUP BY name;

--3- A query that uses Ranking to classify medications based on the number of sales

select trade_name, ROW_NUMBER() over (order by count(sale_id) desc) as ranking, s.Drug_ID, count(sale_id)
from Drug d inner join Sale s
on s.Drug_ID=d.Drug_ID
group by Trade_Name, s.Drug_ID
```

```
-------------------------------------------------------------------------------------------
--1.Create a Rule to restrict the amount of prescribed medications to not exceed a certain amount.
create rule Restrict_Amount
as @Y>=300
sp_bindrule Restrict_Amount , 'Prescription.quantity';


--2- Create a rule to restrict the price of medicines not to exceed a certain price

create rule Restrict_price
as @t<=300
sp_bindrule Restrict_price , 'Prescription.price';

--3- Create a Rule to restrict the history of recipes to not be in the future
create rule prescription_date as @x< getdate()+1
go
sp_bindrule  prescription_date, '[dbo].[prescription].[Date]'


insert into Prescription(Pres_ID,PID,Phys_ID,Drug_ID,Date,Quantity,Price)
values (70,14,4,14,'12-12-2025',40,40)

insert into Prescription(Pres_ID,PID,Phys_ID,Drug_ID,Date,Quantity,Price)
values (70,14,4,14,'12-12-2025',40,40)
```

```sql
---4- Create a Trigger to record modifications to the medication schedule in a separate recording table

create table audit
(
drug_idnew int,
drug_idold int,
trade_namenew varchar(30),
trade_nameold varchar(30),
company_idnew int,
company_idold int
)

create trigger t4

on drug
after update
as
begin
insert into audit ( drug_idnew , drug_idold , trade_namenew , trade_nameold ,company_idnew , company_idold )

select i.Drug_ID, de.Drug_ID, i.Trade_Name, de.Trade_Name, i.Company_ID, de.Company_ID
from deleted de inner join inserted i
on i.Drug_ID=de.Drug_ID
end
```

```sql
--5- Create a Trigger to prevent deleting an employee if his sales are higher than 1000

create trigger t5
on employee
instead of delete
as
if exists (select e.name, sum(quantity * price) as emp_sales from Employee e inner join Pharmacy ph
on e.Phar_ID=ph.Phar_ID
inner join Sale s
on s.Phar_ID=ph.Phar_ID
group by e.Name
having sum(quantity * price) > 1000 )

begin
select 'you can not delete him'
end
else
begin
delete e from Employee e inner join deleted d
on e.Employee_ID=d.Employee_ID
end

delete from Employee
where Name='Logan Flores'
```

```sql
--2- Create a Trigger to send a notification when a new prescription is added

CREATE or alter TRIGGER NotifyNewPrescription
ON Prescription
AFTER INSERT
AS
BEGIN

    select 'New prescription added for patient: '
    END;
INSERT INTO Prescription (Pres_ID, Phys_ID, Drug_ID, Quantity, Price, Date)
VALUES (65,  2, 2, 5, 15.0, '2026-03-16');

--3- Create Trigger to update patient table when patient details are updated

 create or alter TRIGGER autoupdatet_p
on dbo.patient
AFTER update
AS
update P set pid = i.PID , name =i.name , sex=i.sex , address=i.address,
contact_no=i.contact_no,insurance_info=i.insurance_info
from  dbo.Patient p inner join inserted i
on p.PID=i.PID
```

```sql
--1.Create a trigger to update drug stock when a new prescription is added.
ALTER TABLE Drug
ADD quantity INT;

CREATE TRIGGER trg_update_drug_stock
ON Prescription
AFTER insert
AS
BEGIN
    UPDATE Drug
    SET Drug.quantity = Drug.quantity - i.quantity
    FROM inserted i JOIN Prescription p ON p.drug_id = i.drug_id
    WHERE Drug.drug_id = i.drug_id;
END;
```

```sql
-----------------------------------------------------------------------
--7.Create an index on a name column in a Patient table to improve the performance
--of queries that search for the patient's name.
create NONCLUSTERED INDEX idx_Patient_Name
on Patient (Name);
-----------------------------------------------------------
--8.create a view that combines data from the Patient and Prescription tables.
create view PatientPrescriptions as
select p.PID,p.Name, p.Sex, p.Address,p.Contact_No,p.Insurance_Info,
        pr.Pres_ID,pr.Date,pr.Quantity,pr.Price

FROM Patient P JOIN Prescription PR
    ON P.PID=PR.PID;
GO
SELECT *
FROM PatientPrescriptions;
```

```sql
--5- Query for prescriptions issued after a certain date, for example '2026-01-01'

select * from Prescription
where Date> '2026-01-01'

--6- Inquiry to obtain the number of patients registered in the base

select count(*) as total_patient
from Patient

--7- A query to obtain the average price of medicines in the prescription table

select AVG(price) as avg_price
from Prescription

--8- Query to extract manufacturer details of medicines bearing a particular Trade_Name, eg 'Ibuprofen'

select m.*
from Drug_Manufacturer m inner join Drug d
on m.Company_ID=d.Company_ID
where Trade_Name='Ibuprofen'
```

```sql
--1- A query to extract all patient names

select name from patient

--2- Inquire to obtain details of all doctors who specialize in “Dermatology”

select * from Doctor
where specialty = 'Dermatology'

--3- Inquiry to obtain the names and prices of medicines from the prescription table

select trade_name, price
from Drug d inner join Prescription p
on d.Drug_ID=p.Drug_ID

--4- A query to extract all employees who work in a particular pharmacy, for example Phar_ID = 2

select e.* from Employee e inner join Pharmacy ph
on e.Phar_ID=ph.Phar_ID
where ph.Phar_ID=2
```

```sql
--13- Inquiry to obtain the name of the patient who has the largest number of prescriptions

select top 1 Patient.name
from Patient inner join Prescription p
on Patient.PID=p.PID
group by Patient.name
order by COUNT(p.Pres_ID) desc

--14- A query to obtain details of prescriptions issued by a specific doctor in a specific period of time

select *
from Prescription
where Phys_ID =3 and Date between '2025-03-01' and '2026-06-01'

--15- Inquiry to obtain the names of medications that have not been filled in any prescription

select trade_name, d.Drug_ID
from Drug d left join Prescription p
on d.Drug_ID=p.Drug_ID
where p.Drug_ID is null

--16- Inquiry to obtain the total quantities required for all medications
```

```sql
--9- Inquiry to obtain the number of doctors who specialize in “Cardiology”

select count(*) from Doctor
where specialty='Cardiology'

--10- Inquiry to obtain the names of patients who do not have health insurance (null insurance_info)
select name
from Patient
where insurance_info is null

--11- Inquiry to get details of all prescriptions whose prices exceed 100

select *
from Prescription
where Price > 100

--12- Query to get the names of doctors who prescribed a particular drug, for example Drug_ID = 12

select dr.d_name
from Doctor dr inner join Prescription p
on dr.Phys_ID=p.Phys_ID
where p.Drug_ID=9
```

```sql
--24- Inquiry to obtain details of patients who have more than one prescription on a specific date

select Patient.*
from Patient inner join Prescription p
on Patient.PID=p.PID
where p.Date='2025-08-01'
group by Patient.address, Patient.contact_no, Patient.insurance_info, Patient.name, Patient.PID, Patient.sex
having COUNT(Pres_ID) > 1

--25- Inquiry to obtain details of doctors who prescribed a specific medication in more than one pharmacy

select dr.*
from Doctor dr inner join Prescription p
on dr.Phys_ID=p.Phys_ID
inner join Drug d
on d.Drug_ID=p.Drug_ID
inner join Sale s
on s.Drug_ID=d.Drug_ID
group by dr.D_name, dr.Phys_ID, dr.specialty
having COUNT(s.Drug_ID) > 1
```

```sql
--16- Inquiry to obtain the total quantities required for all medications

select trade_name, sum(quantity) as total_quantity
from Prescription p inner join Drug d
on p.Drug_ID=d.Drug_ID
group by Trade_Name

--17- Inquiry to obtain the name of the manufacturer that manufactures the largest number of medicines

select top 1 dm.name
from Drug_Manufacturer dm inner join Drug d
on dm.Company_ID=d.Company_ID
group by dm.Name
order by count(d.Drug_ID) desc


--18- A query to get the number of prescriptions issued in each month of a given year, for example 2026

select MONTH(date) as month, count(pres_id)
from Prescription
where year(Date) = 2026
group by month(Date)
```

```sql
--19- Inquiry to obtain details of the pharmacy that has the largest number of employees

select top 1 ph.*, COUNT(e.Phar_ID)
from Pharmacy ph inner join Employee e
on ph.Phar_ID=e.Phar_ID
group by ph.Phar_ID, ph.Address, ph.Fax, ph.Name
order by COUNT(e.Phar_ID) desc

--20- Inquiry to obtain the names of medicines whose prices exceed the general average

select trade_name
from Drug d inner join Prescription p
on d.Drug_ID=p.Drug_ID
where Price > (select avg(price) from Prescription )

--21- A query to get details of prescriptions containing a certain amount of medicine, for example 10

select *
from Prescription
where Quantity=10
```

```sql
--22- A query to obtain the names of drugs manufactured by specific companies, for example companies with names s

select d.Trade_Name
from Drug d inner join Drug_Manufacturer dm
on d.Company_ID=dm.Company_ID
where dm.Name like 'p%'

--23- Inquiry to obtain details of medications that have been prescribed more than 10 times

select Trade_Name
from Drug d inner join Prescription p
on d.Drug_ID=p.Drug_ID
group by d.Trade_Name
having COUNT(p.drug_id) > 3
--------------------------------------------------------------------------------
SELECT Drug.Trade_Name, COUNT(Prescription.Pres_ID) AS prescription_count
FROM Drug
JOIN Prescription ON Drug.Drug_ID = Prescription.Drug_ID
GROUP BY Drug.Trade_Name
HAVING COUNT(Prescription.Pres_ID) > 10
```